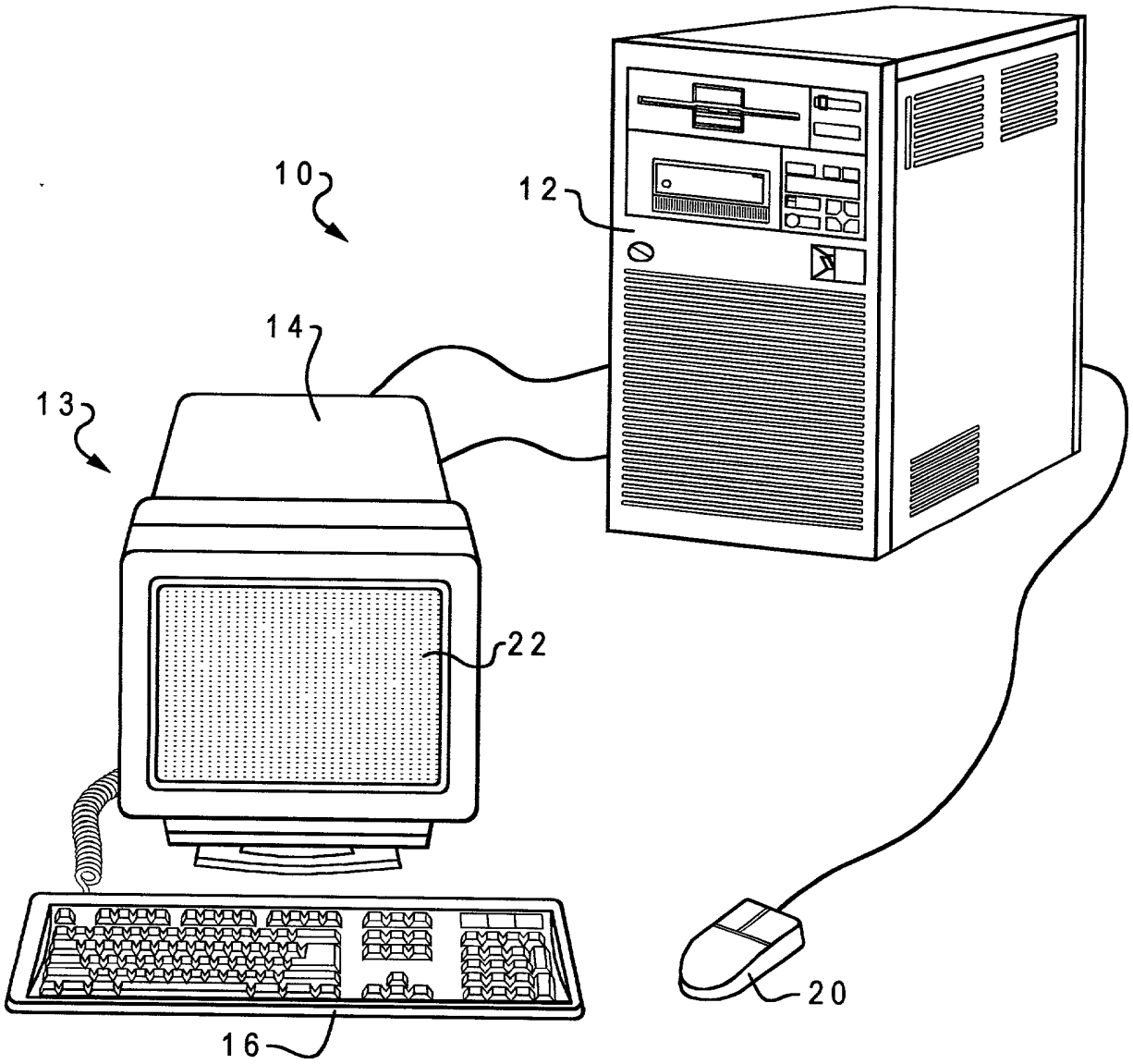

Fig. 1

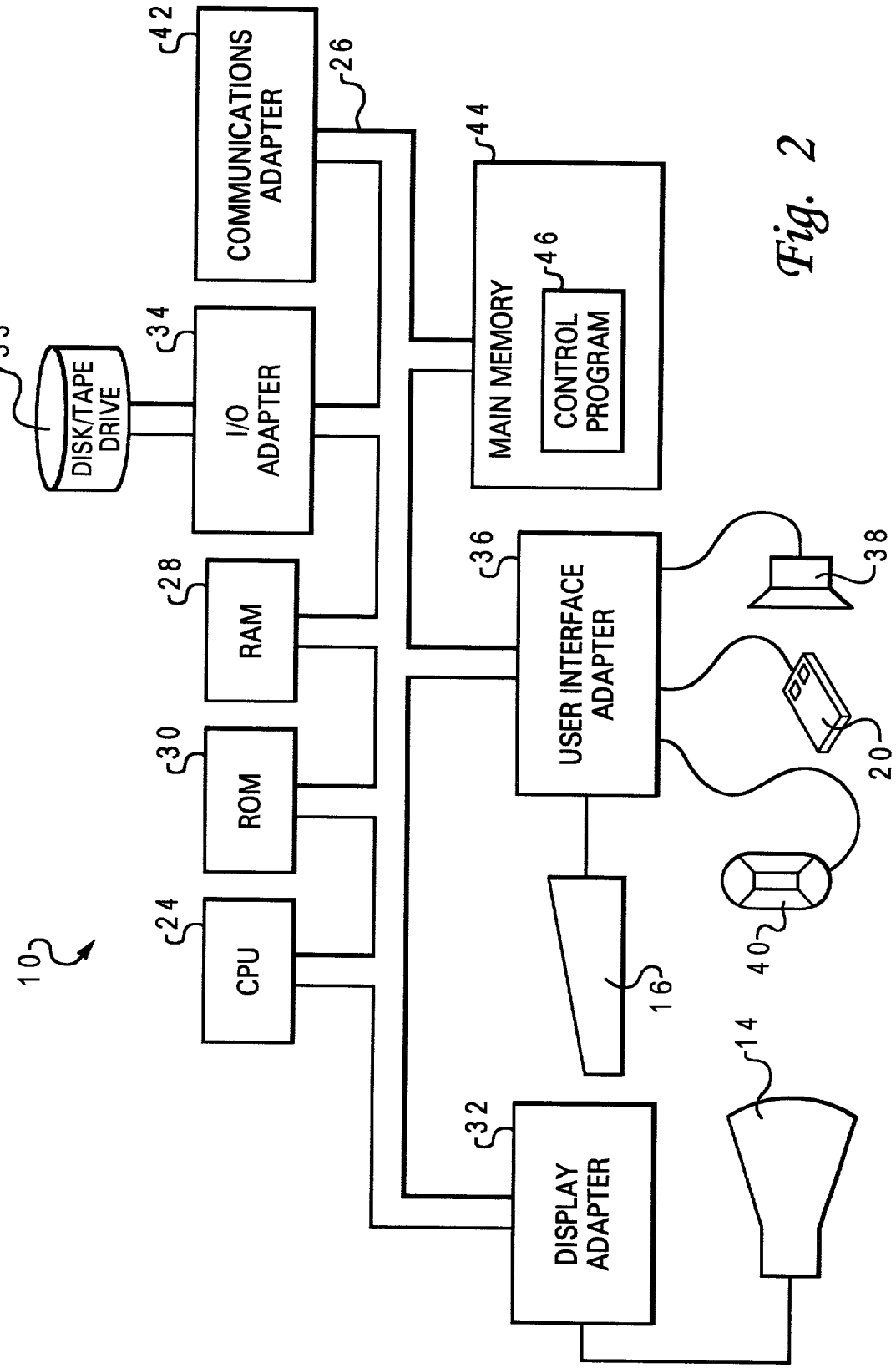
10
CPU
24
ROM
30
RAM
28
I/O ADAPTER
34
DISK/TAPE DRIVE
33
COMMUNICATIONS ADAPTER
42
26
MAIN MEMORY
44
CONTROL PROGRAM
46
USER INTERFACE ADAPTER
36
DISPLAY ADAPTER
32
16
14
40
20
38

Fig. 2

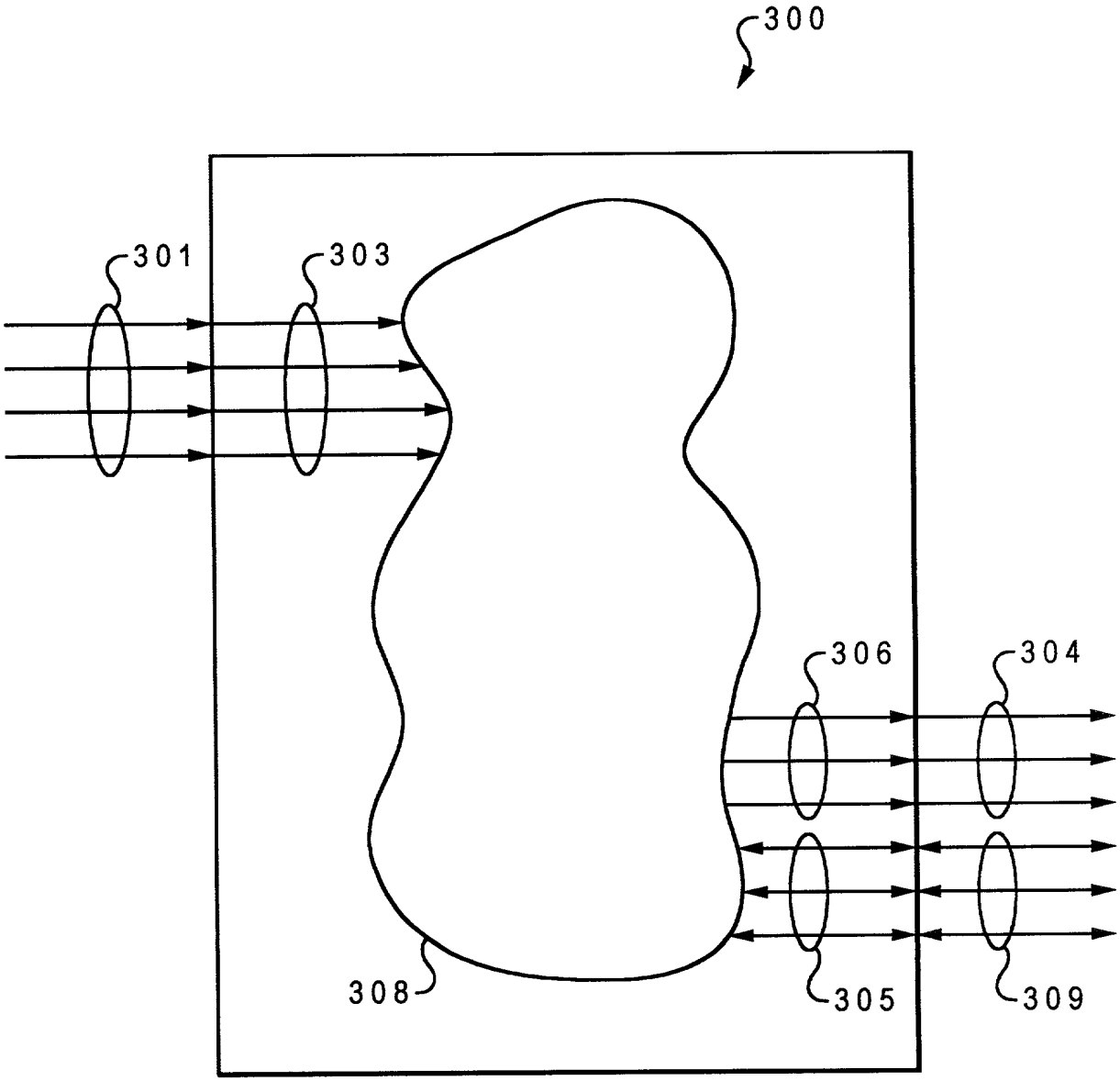

Fig. 3A

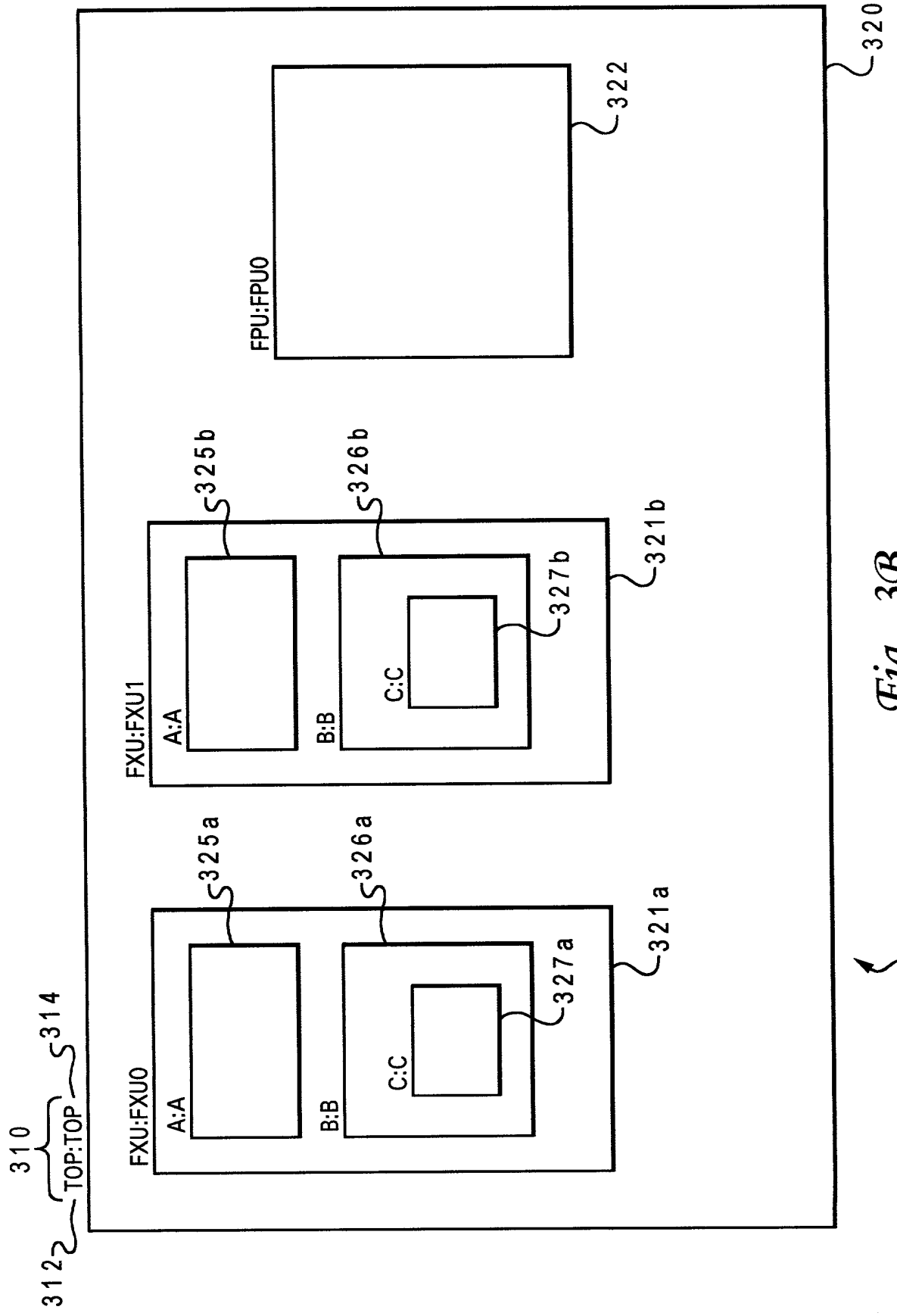
310
TOP:TOP
314
312
FXU:FXU0
A:A
B:B
C:C
325a
326a
327a
321a
FXU:FXU1
A:A
B:B
C:C
325b
326b
327b
321b
FPU:FPU0
322
320
329

Fig. 3B

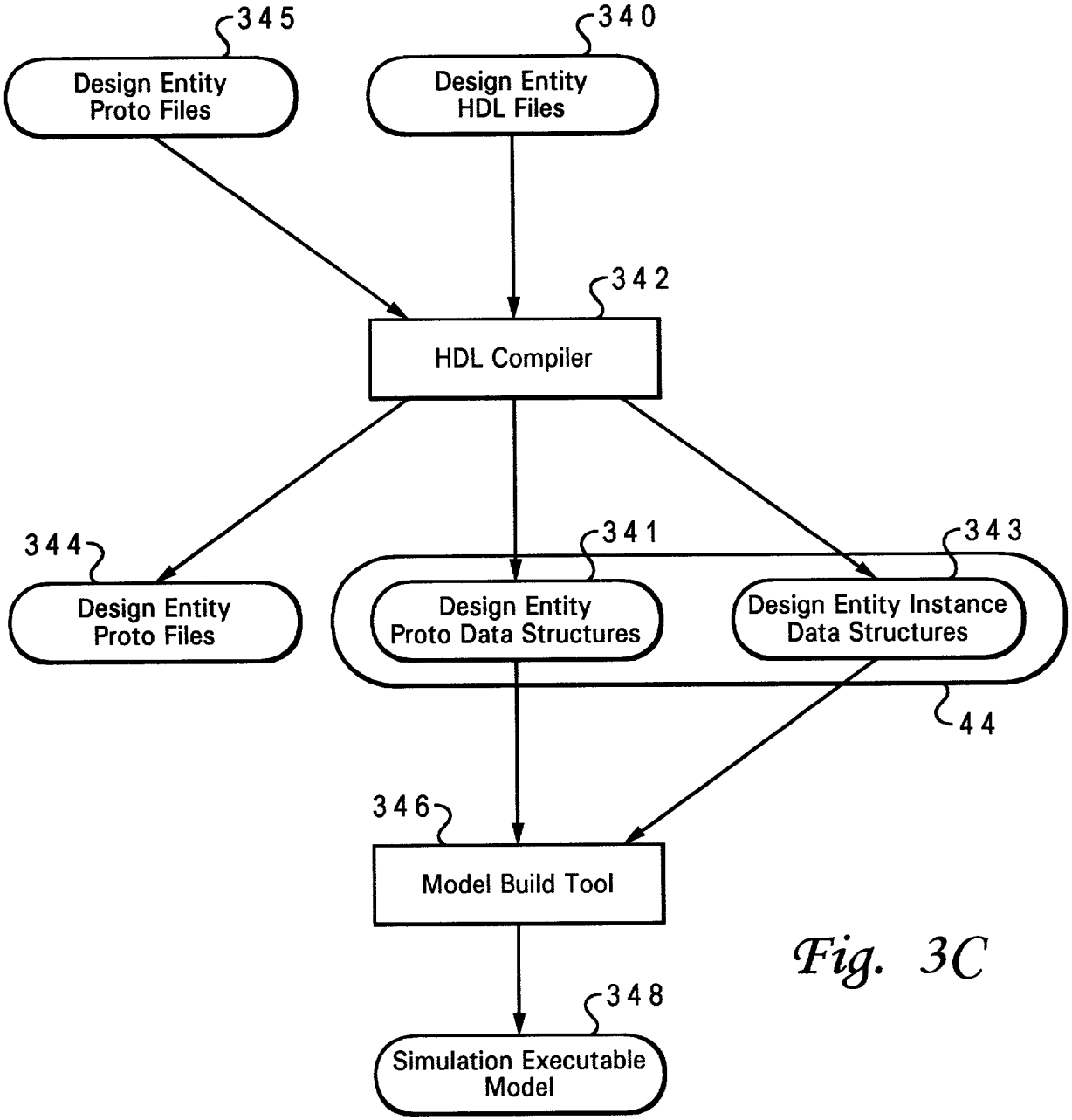
345
Design Entity Proto Files
340
Design Entity HDL Files
342
HDL Compiler
344
Design Entity Proto Files
341
Design Entity Proto Data Structures
343
Design Entity Instance Data Structures
44
346
Model Build Tool
348
Simulation Executable Model

Fig. 3C

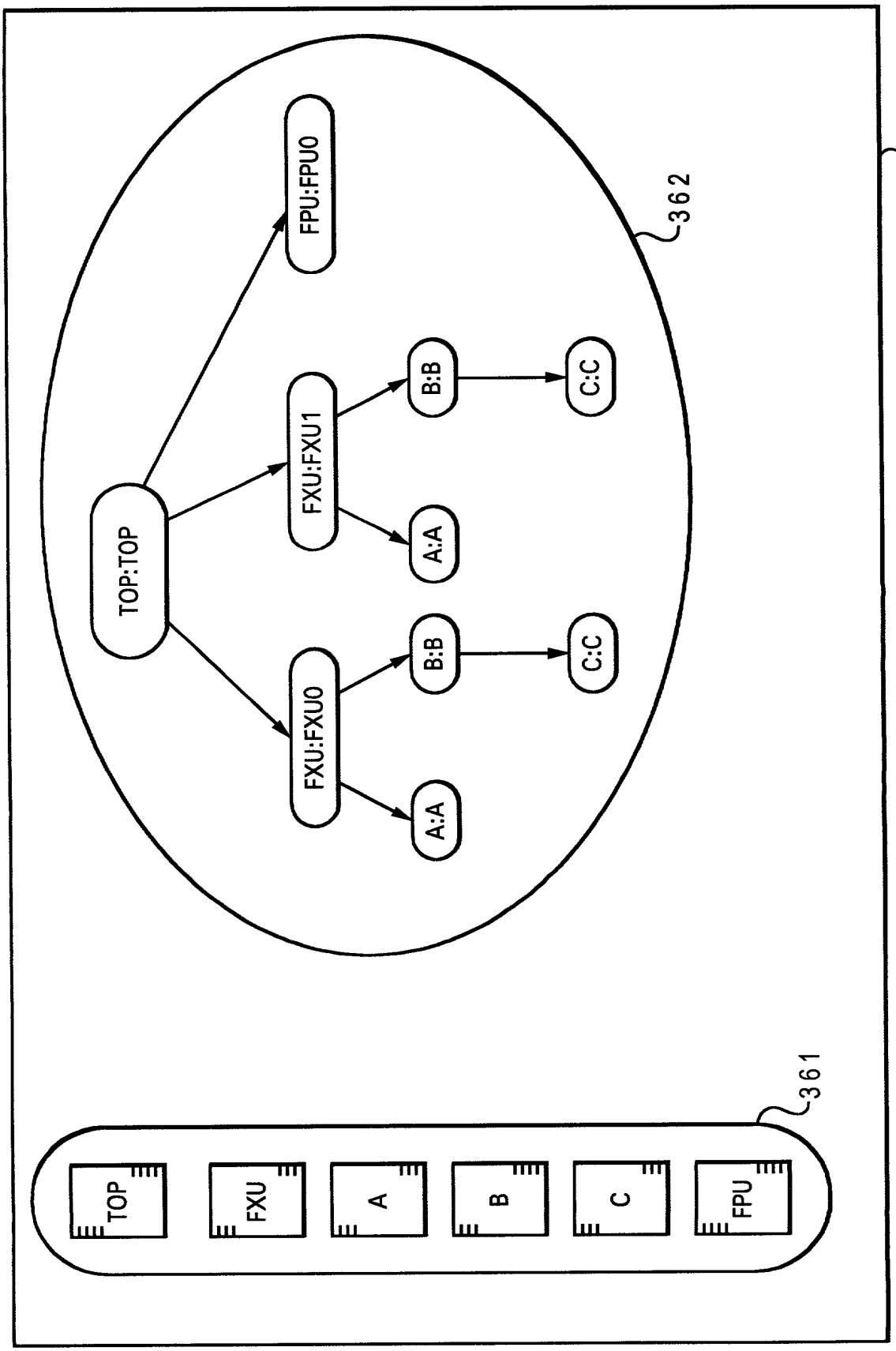
TOP
FXU
A
B
C
FPU
361
TOP:TOP
FPU:FPU0
FXU:FXU1
FXU:FXU0
A:A
B:B
C:C
A:A
B:B
C:C
362
44

Fig. 3D

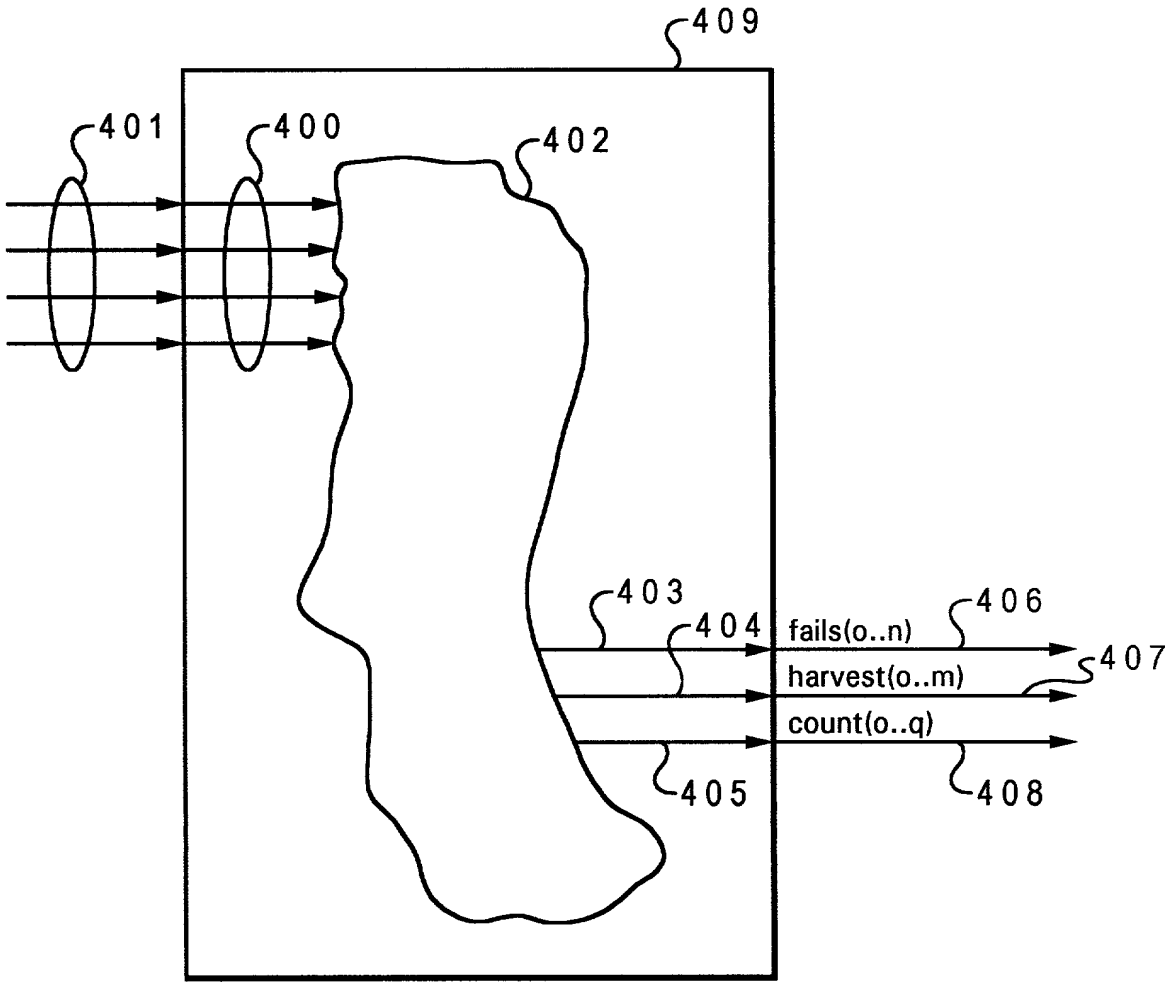
409
401
400
402
403
404
fails(o..n)
406
407
harvest(o..m)
count(o..q)
405
408

Fig. 4A

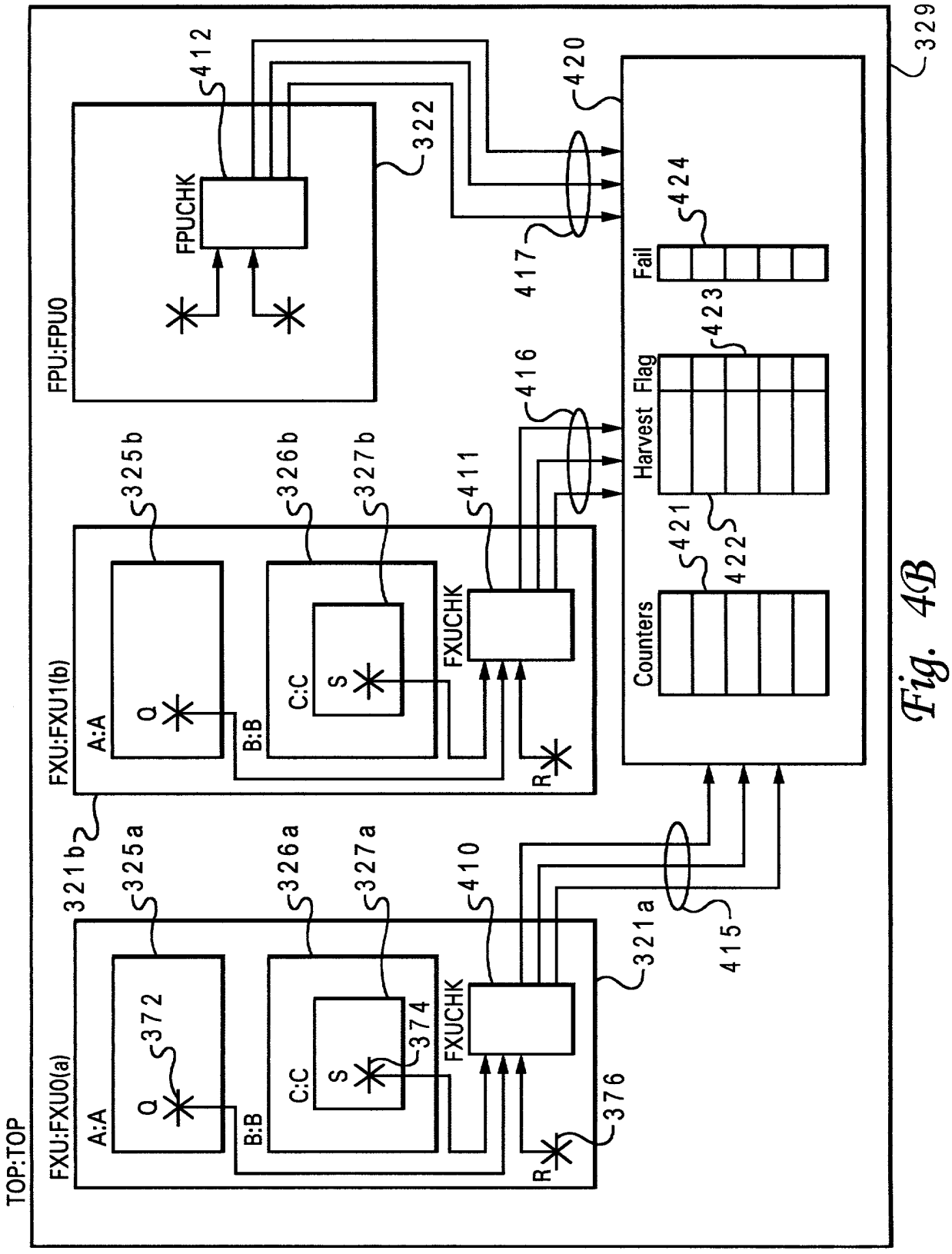
TOP:TOP
FXU:FXU0(a)
A:A
Q
372
B:B
C:C
S
374
FXUCHK
376
R
321b
325a
326a
327a
410
321a
415
FXU:FXU1(b)
325b
326b
327b
411
416
FPU:FPU0
412
FPUCHK
322
417
420
329
Fail
424
Harvest Flag
423
Counters
421
422

Fig. 4B

```
ENTITY FXUCHK IS                                                    } 450

    PORT(   S_IN        :       IN std_ulogic;
            Q_IN        :       IN std_ulogic;
            R_IN        :       IN std_ulogic;
            clock       :       IN std_ulogic;
            fails       :       OUT std_ulogic_vector(0 to 1);
            counts      :       OUT std_ulogic_vector(0 to 2);
            harvests    :       OUT std_ulogic_vector(0 to 1);
        );

452 {   --!! BEGIN
        --!! Design Entity: FXU;

453 {   --!! Inputs
        --!! S_IN       = >     B.C.S;
        --!! Q_IN       = >     A.Q;
        --!! R_IN       = >     R;
        --!! CLOCK      = >     clock;
        --!! End Inputs

454 {   --!! Fail Outputs;
        --!! 0 : "Fail message for failure event 0";
        --!! 1 : "Fail message for failure event 1";
        --!! End Fail Outputs;

455 {   --!! Count Outputs;
        --!! 0 : <event0> clock;
        --!! 1 : <event1> clock;
        --!! 2 : <event2> clock;
        --!! End Count Outputs;

456 {   --!! Harvest Outputs;
        --!! 0 : "Message for harvest event 0";
        --!! 1 : "Message for harvest event 1";
        --!! End Harvest Outputs;

457 {   --!! End;                                                    } 451

    ARCHITECTURE example of FXUCHK IS

    BEGIN

        ... HDL code for entity body section ...                    } 458

    END;                                                            } 440
```

Fig. 4C

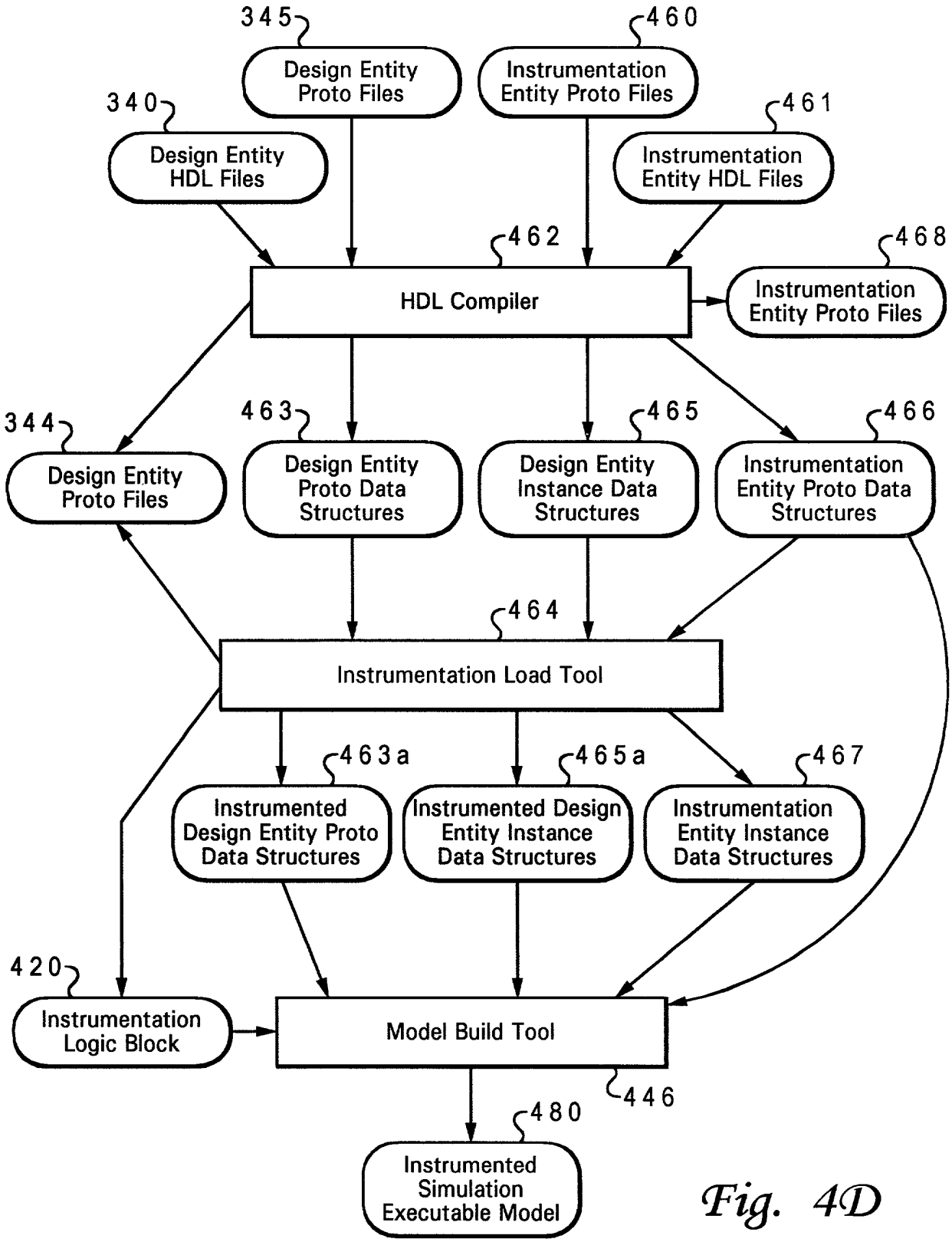
345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

Fig. 4D

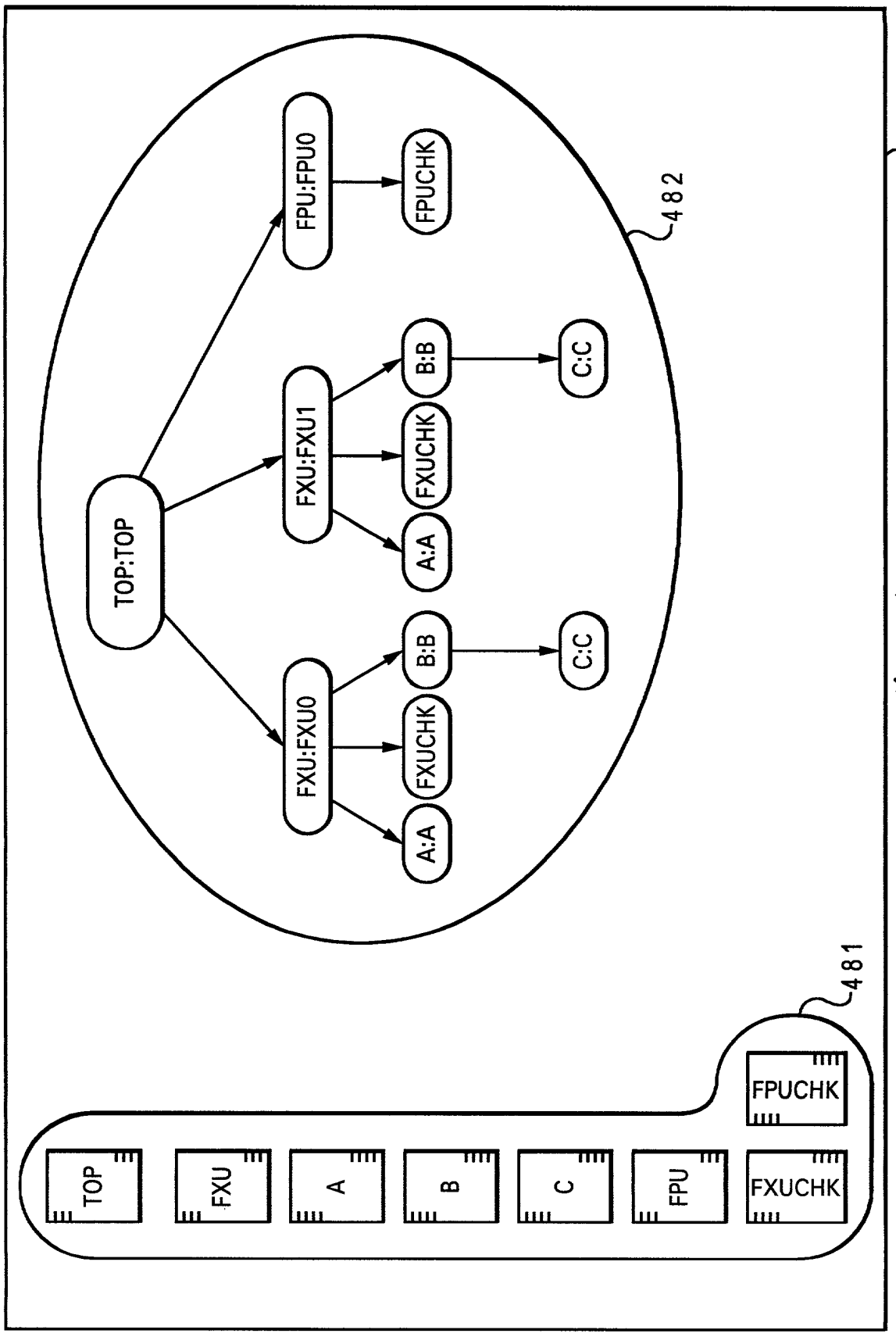
TOP
FXU
A
B
C
FPU
FXUCHK
FPUCHK
481
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
C:C
FPUCHK
482
44

Fig. 4E

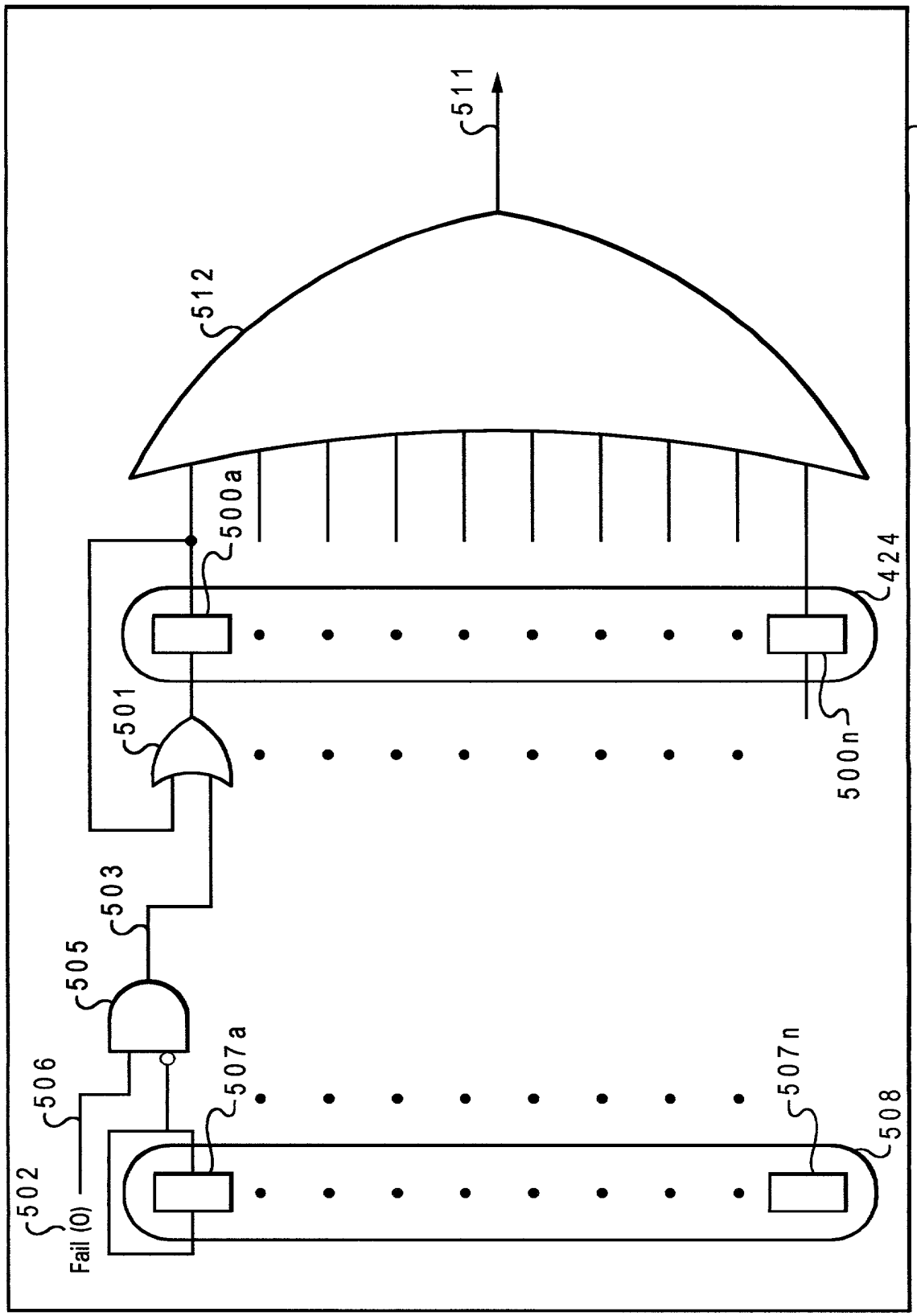
511
420
512
500a
424
501
500n
503
505
506
507a
507n
502
Fail (0)
508

Fig. 5A

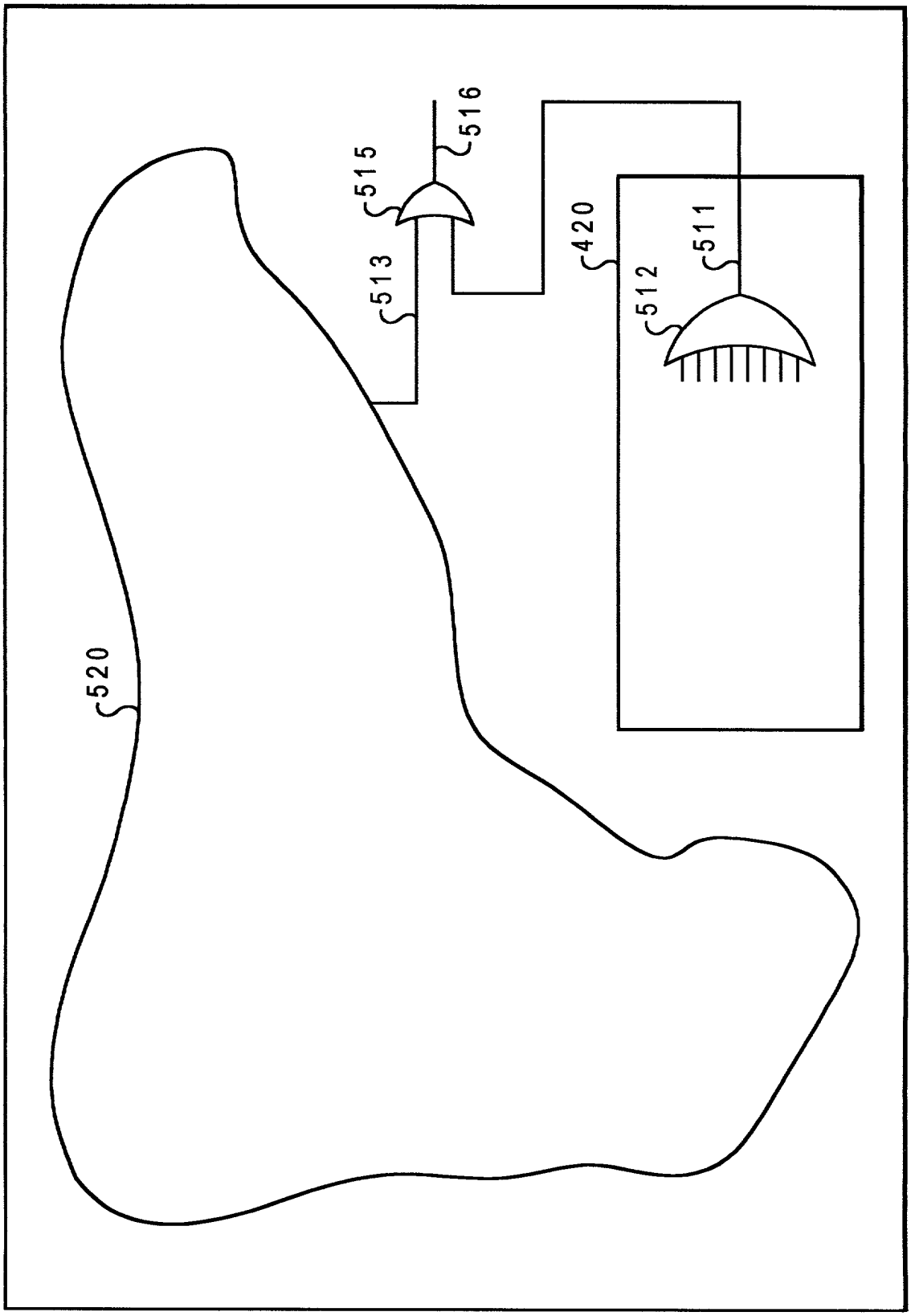

Fig. 5B

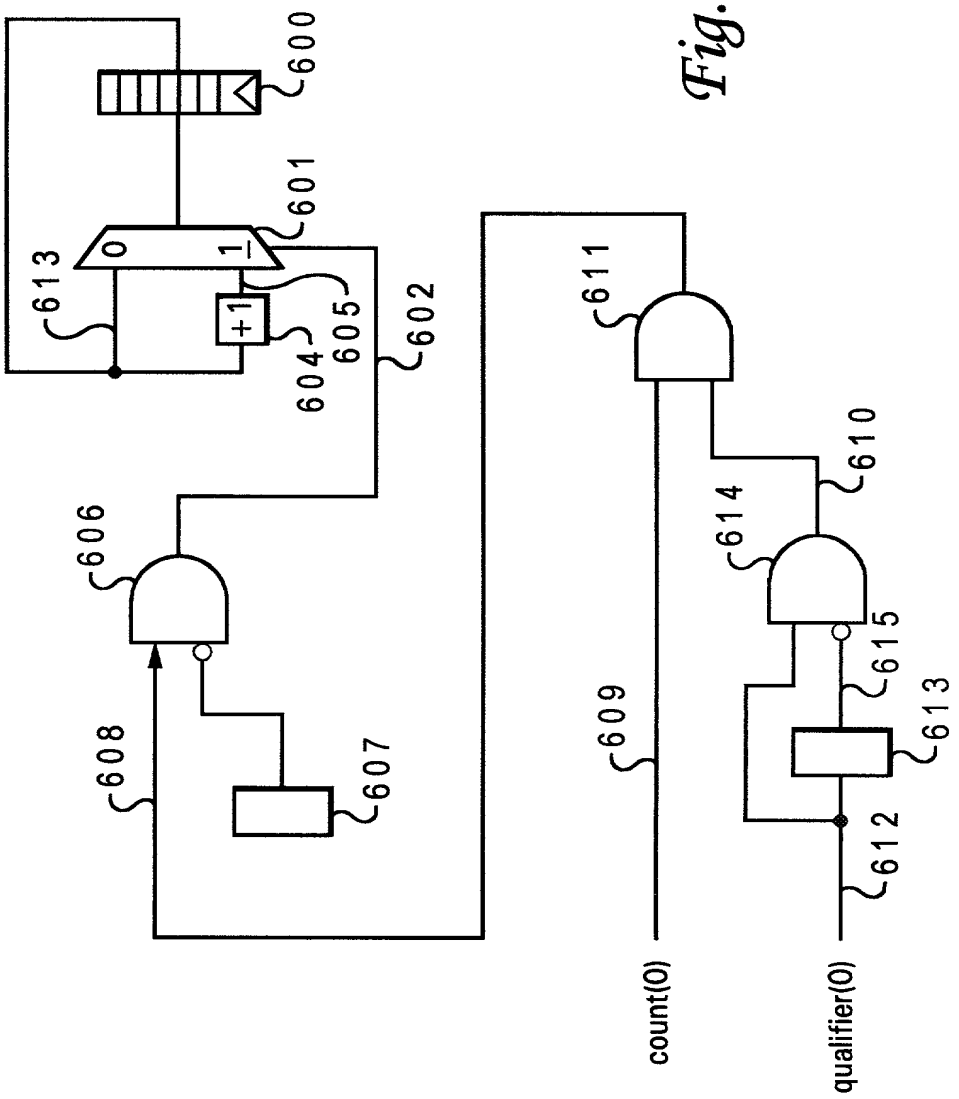
600
601
613
0
1
+1
604
605
602
606
608
607
611
609
614
610
615
613
612
count(0)
qualifier(0)

Fig. 6A

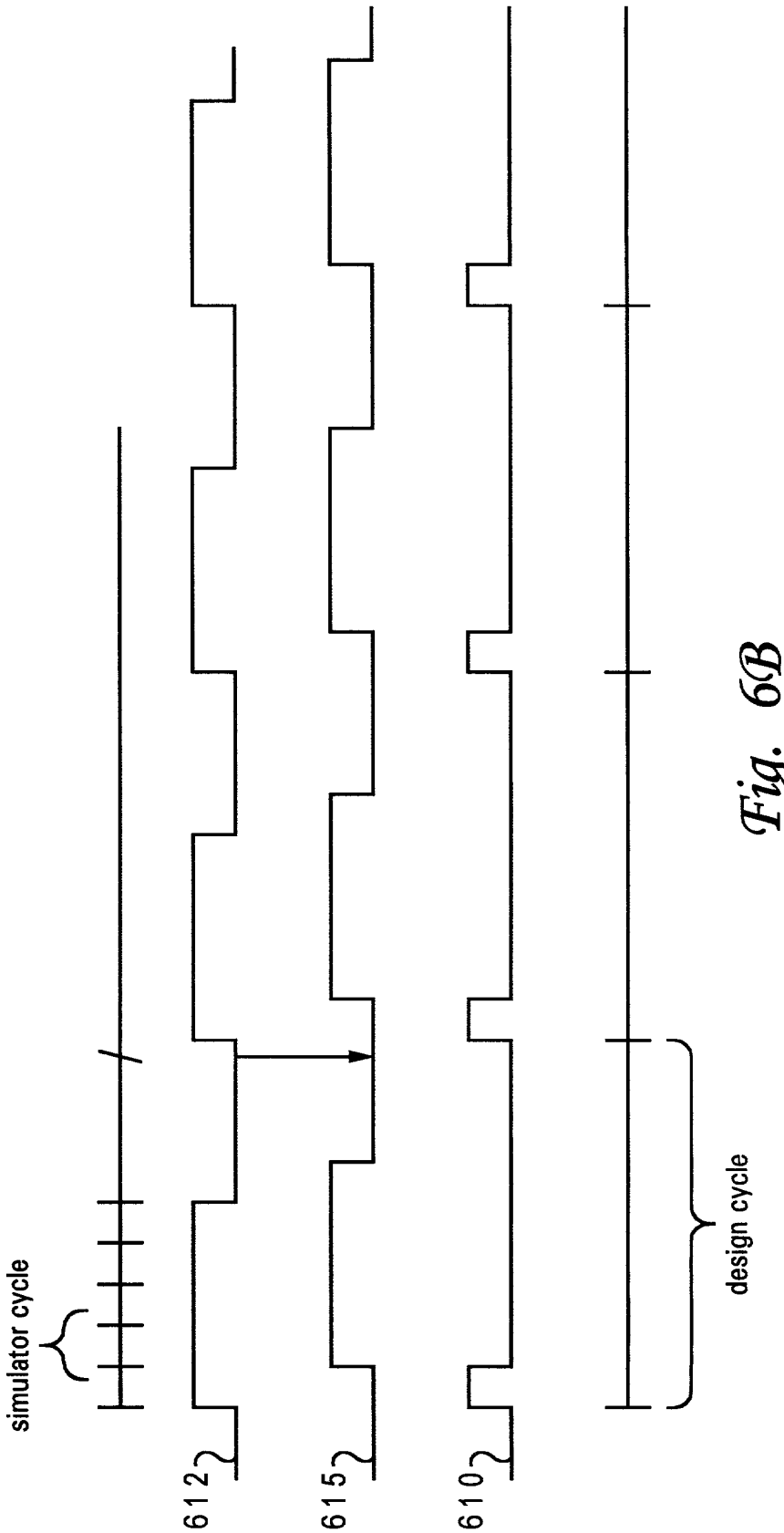

Fig. 6B

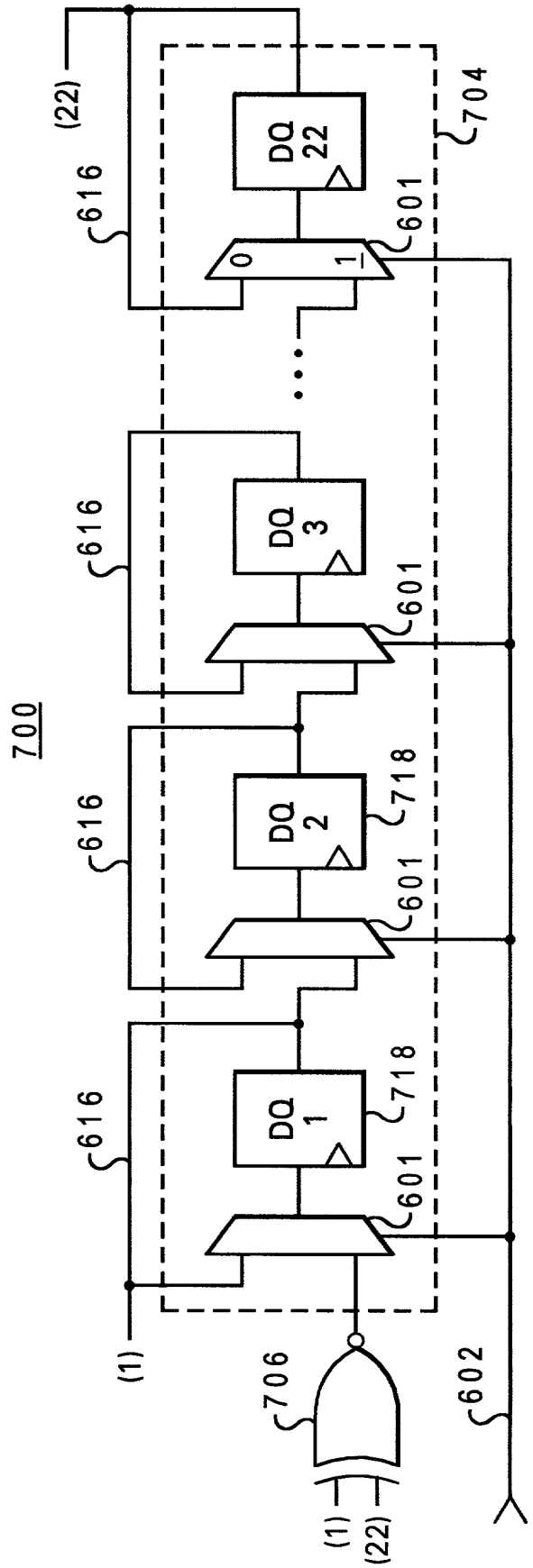

Fig. 7

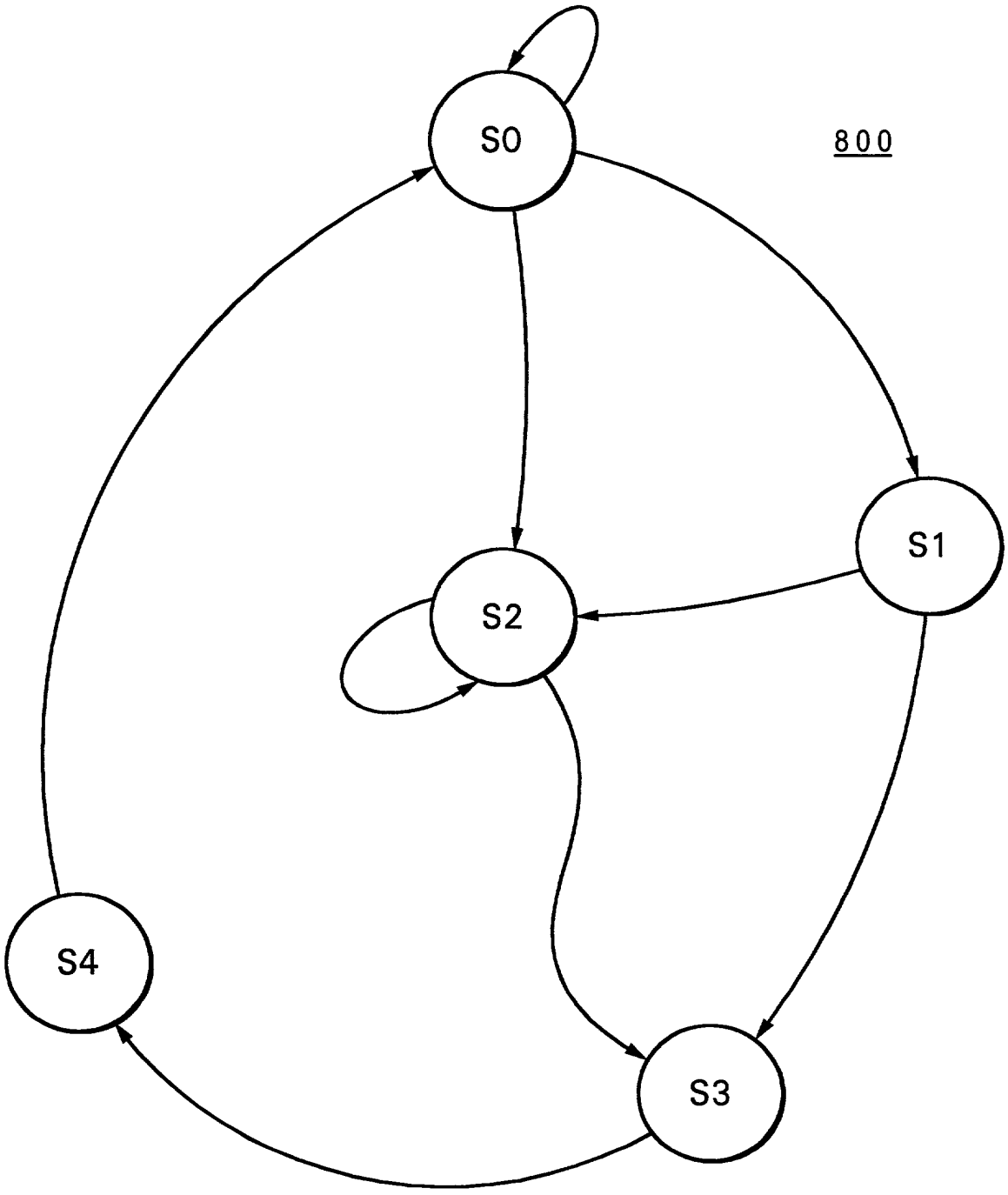

Fig. 8A
Prior Art

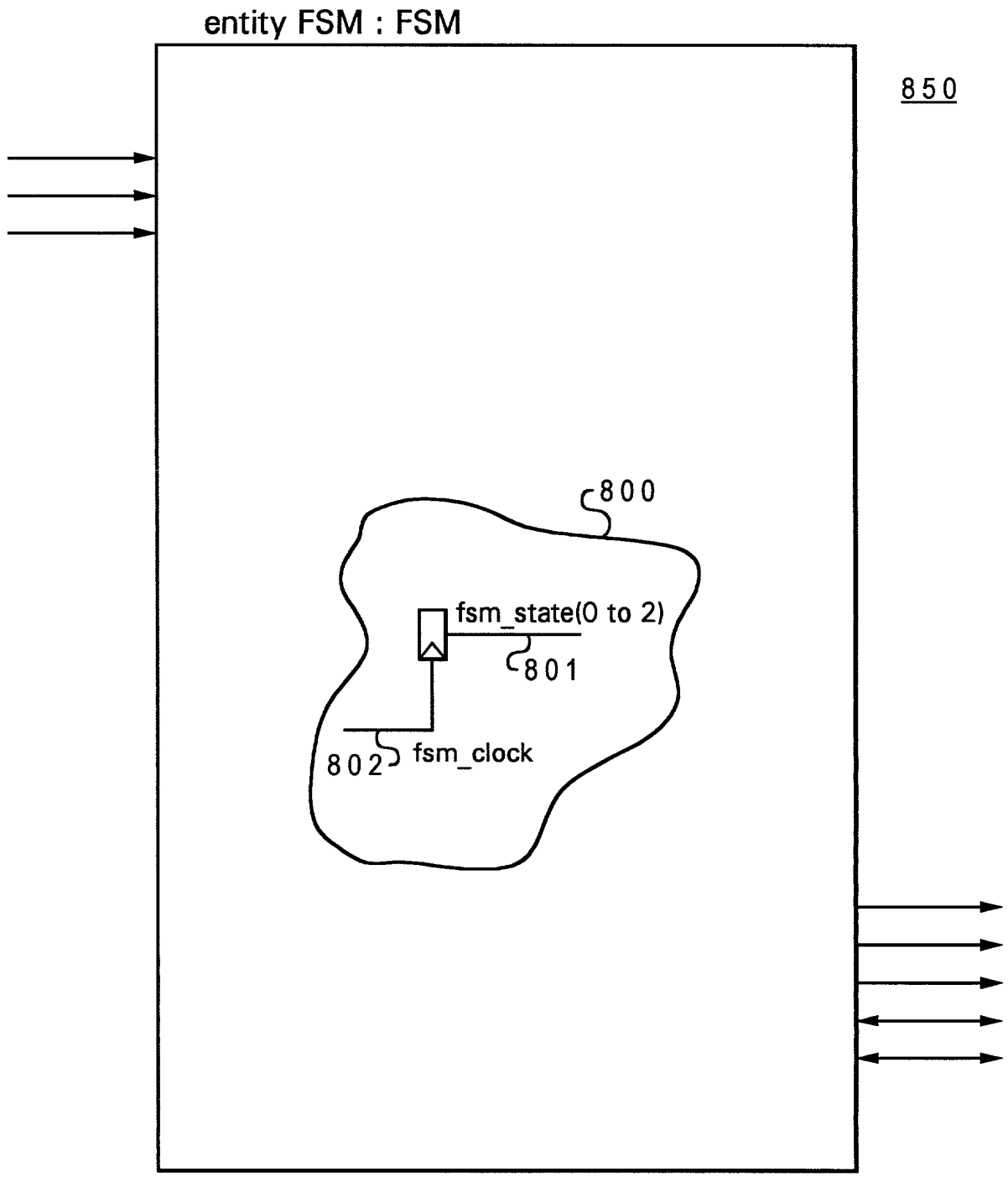

Fig. 8B
Prior Art

```
ENTITY FSM IS

    PORT(
            ....ports for entity fsm....
        );

    ARCHITECTURE FSM OF FSM IS

    BEGIN

            ... HDL code for FSM and rest of the entity ...

            fsm_state(0 to 2) <= ... Signal 801 ...

853 --!! Embedded FSM : examplefsm;
859 --!! clock        : (fsm_clock);
854 --!! state_vector : (fsm_state(0 to 2));
855 --!! states       : (S0, S1, S2, S3, S4);
856 --!! state_encoding : ('000', '001', '010', '011', '100');
    --!! arcs         : (S0 => S0, S0 => S1, S0 => S2,
857 --!!                 (S1 => S2, S1 => S3, S2 => S2,
    --!!                 (S2 => S3, S3 => S4, S4 => S0);
858 --!! End FSM;

                ,
                ,
                ,
                ,
                ,
                ,
                ,

    END;
```

852 860

Fig. 8C

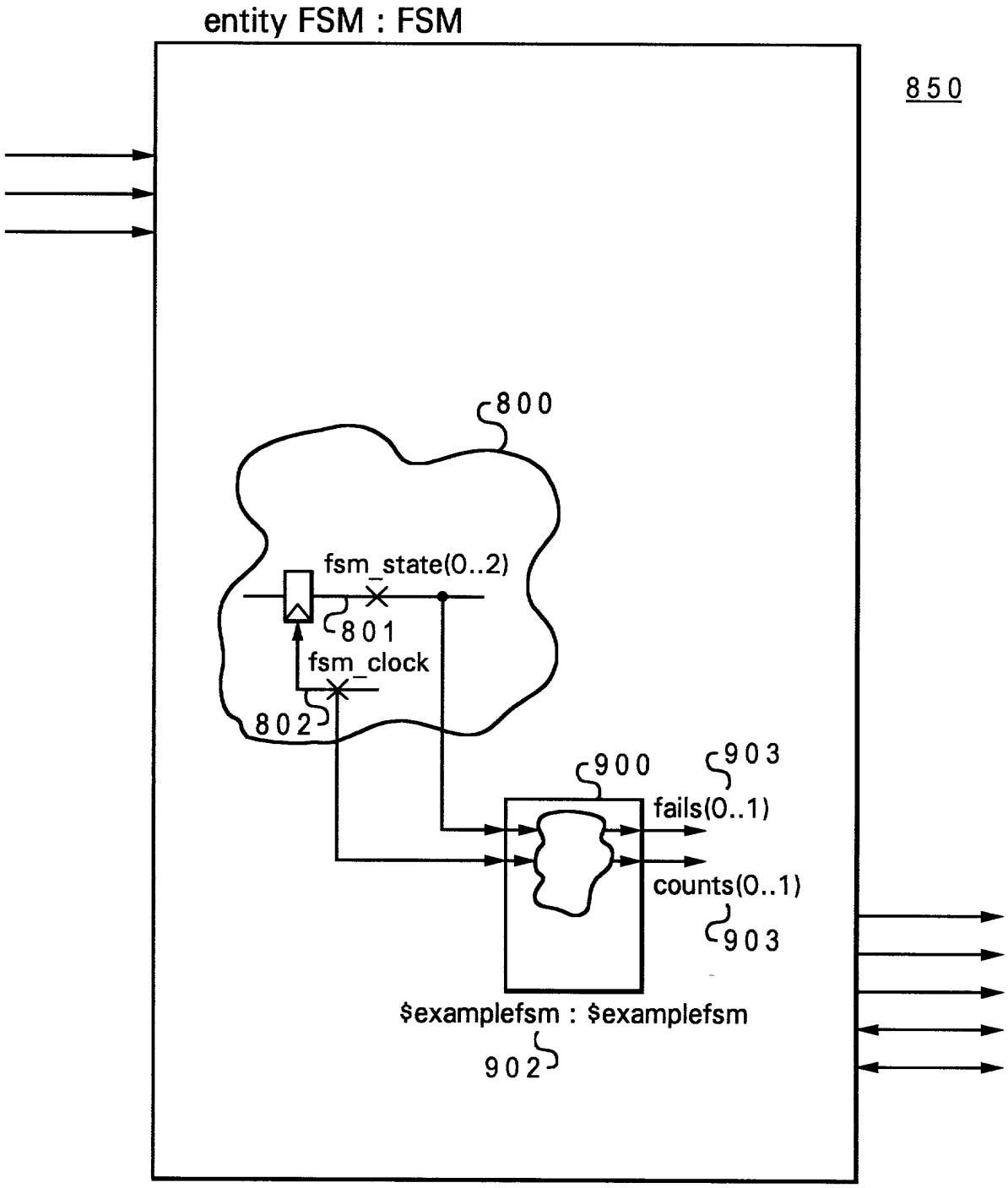
entity FSM : FSM
850
800
fsm_state(0..2)
801
fsm_clock
802
900
903
fails(0..1)
counts(0..1)
903
$examplefsm : $examplefsm
902

Fig. 9